Experiments in Electronic Plagiarism Detection
Computer Science Department, TR 388

Caroline Lyon, Ruth Barrett, James Malcolm

1. Introduction

This report first describes three plagiarism detection systems: the Ferret, developed here at the University of Hertfordshire, (Section 2), Turnitin (Section 3) and CopyCatch (Section 4). The Ferret is a stand-alone system that can be installed on any PC and run on student work by naïve users. Turnitin is a remote system, to which student work is submitted electronically. Turnitin compares each file with Web material and with files held in a database of previously submitted work. CopyCatch, like Ferret, is a local system that can be easily installed on any PC. Our experiments are conducted on the freely distributed basic version.

JISC has conducted a survey of plagiarism detection systems [Bull 2001], but the Ferret was not included because it was not a commercially available product at the time the survey started. The use of Turnitin is supported by JISC (the Joint Information Systems Committee).

In Section 5 experiments comparing the performance of Ferret with Turnitin and CopyCatch are described. Functionality, usability, ease of interpreting results, scope and robustness are all investigated. Section 6 reports on field trials with Ferret, in Dutch as well as in English.

Section 7 gives our conclusions and describes future plans. In summary, we find that Ferret complements Turnitin. It effectively detects collusion (similar passages of text) in work done by large numbers of students, displaying the potential collusion in a more accessible way than Turnitin. However, Turnitin is effective in detecting potential plagiarism from Web sources. Ferret compares student work with some Web material, but not the very large quantity that Turnitin can access. In practice Ferret on a local machine is simpler to use than Turnitin, particularly as submitting a large number of files to Turnitin is prohibitively time-consuming. CopyCatch detects collusion but is not very effective. Like Ferret it is an easy to use local system and has similar aims to Ferret, and it can detect gross collusion. However, it fails to find some obvious examples and, on the other hand, marks as suspiciously similar documents which were independently written: it produces both false negatives and false positives, as described below. In all these systems, the whole issue of electronic submission, necessary for them all, was not always straightforward.

2. The Ferret

The Ferret software has been developed in the Computer Science department. The detection algorithm was devised by Caroline Lyon and coded by Bob Dickerson and James Malcolm. The associated software, in various versions, was written by several members of the department.

2.1 Theoretical basis

The principle underlying the operation of the Ferret is that a piece of text can be characterised by a set of small features: word trigrams. A trigram is a sequence of 3 adjacent words, so the following string

plagiarism is a common problem

can be decomposed into a set of overlapping trigrams:

plagiarism is a *is a common* *a common problem*

The phenomenon we exploit is the fact that common trigrams constitute a very small proportion of the trigrams in independent texts. This is a consequence of the distribution of words, as explained in [Lyon *et al*. 2001]. If the same subject matter is reported by independent writers there will typically be few matching trigrams as a proportion of the whole, as has been empirically demonstrated [Gibbon *et al*. 1997]. This is the same phenomenon that creates a fundamental problem in automated speech recognition: the sparse data issue [Ney *et al*. 1991]; but what is a problem in speech recognition is here exploited for the detection of plagiarism and collusion.

The core of the Ferret system is a program to convert a piece of text to a set of trigrams, and then compare it with sets of trigrams from other texts. If the number of matches between two texts exceeds a threshold, then they are suspiciously similar, and can be displayed side by side with matching passages highlighted. The texts do not have to have identical passages to be detected: if some words are deleted, inserted or substituted similar passages can still be picked up.

Independently written texts, even if they are written by the same author on the same subject (but on different occasions), do not usually have many common trigrams, as observed in empirical investigations (see Section 5).

The similarity between files is based on a set theoretic measure "Resemblance", *R* [Broder 1998]. If N_A is the set of trigrams in file A and N_B the set of trigrams in file B, then

$$R = \frac{N_A \cap N_B}{N_A \cup N_B}$$

This is also known as the Jaccard coefficient [Manning 1999, page 29]

Similarity could also be measured by the number of matching trigrams as a proportion of all distinct trigrams in the file.

$$S = \frac{\text{number of matching trigrams}}{\text{number of distinct trigrams}} * 100\ \%$$

In order to understand the significance of these measures, we need to observe the range of values for independently written, non plagiarised texts, and find thresholds above which copying is suspected, as described in Section 5.

The core program for trigram matching was developed in C++ on Unix. It can be used as it is for copy detection in large document collections, *e.g*. 1 million news reports [Lyon *et al.* 2002], as well as for the applications described in this report.

In an educational environment, large numbers of students' assignments are submitted electronically, and each compared with each for similarity that indicates plagiarism or collusion. Thus 300 essays would involve (300 * 299) / 2 = 44,850 comparisons.

2.1.1 Note on counting number of words in a file: Using the word count facility in Microsoft Word, or the Unix command wc, any string of characters separated by white space, is considered a word. In Ferret there must be a letter in the string for it to count as a word. Numerals and standalone punctuation marks are ignored. Thus the word count of a file in Ferret may not exactly match the Word or Unix measure.

2.2 Using the Ferret

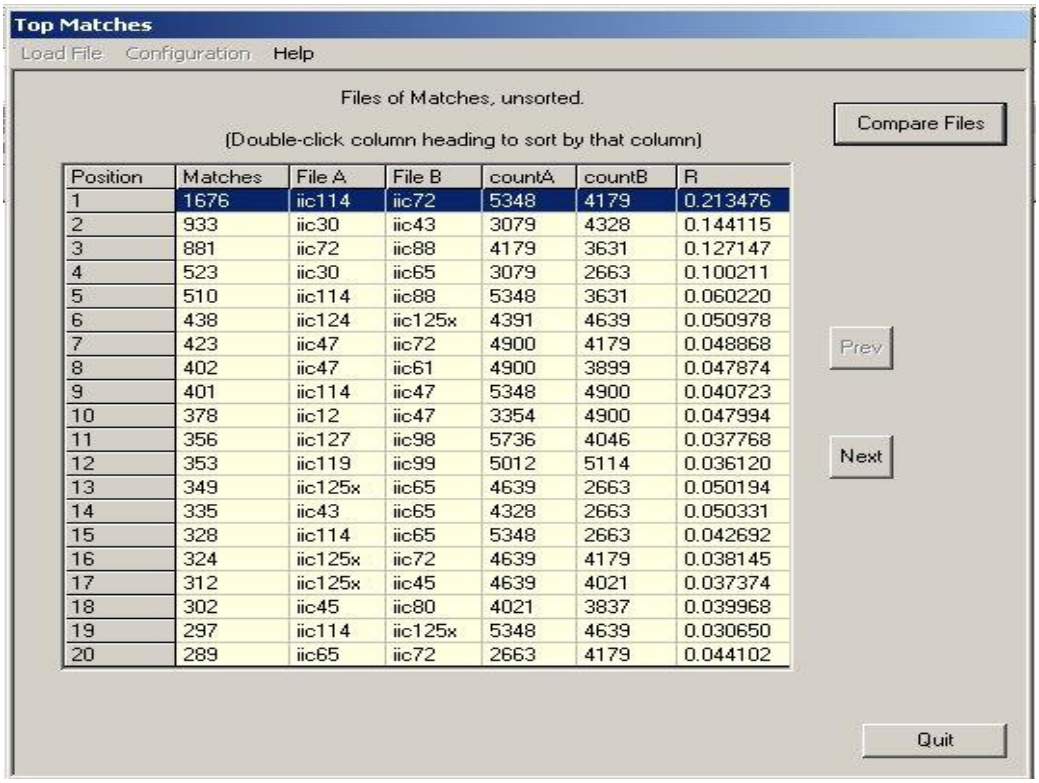
For practical use by naïve users, the program was moved onto a Windows platform, converted to Visual C++, and nicely packaged. The Ferret package has 3 stages: the preliminary format conversion, the trigram matching program, and the final display facility.

A preliminary stage in Visual Basic was added, so that student work could be automatically converted to the necessary text format, prior to processing. Figures and images are ignored. Lecturers have to collect in students' assignments in electronic form, in either Word format (.doc or .rtf), or text (.txt) (or shortly .pdf), put them in a folder, then run the Ferret package.

Any files submitted in the wrong format are reported. In this case the format may be corrected (*e.g.* if the student has named the file with the suffix .doc.doc) or the file can be rejected (*e.g.* if the student has added the suffix .doc to a file which is not in Word format).

Material from the Web can be semi-automatically added to students' work, for comparison. The lecturer has to submit a set of keywords for searching, then 50 -- 100 pages at a time can be downloaded, converted to the correct format and added to the set of students' assignments.

A front end was developed, also in Visual C++; a table shows level of similarity between each pair of documents, in ranked order, as shown in Figure 1.



Position	Matches	File A	File B	countA	countB	R
1	1676	iic114	iic72	5348	4179	0.213476
2	933	iic30	iic43	3079	4328	0.144115
3	881	iic72	iic88	4179	3631	0.127147
4	523	iic30	iic65	3079	2663	0.100211
5	510	iic114	iic88	5348	3631	0.060220
6	438	iic124	iic125x	4391	4639	0.050978
7	423	iic47	iic72	4900	4179	0.048868
8	402	iic47	iic61	4900	3899	0.047874
9	401	iic114	iic47	5348	4900	0.040723
10	378	iic12	iic47	3354	4900	0.047994
11	356	iic127	iic98	5736	4046	0.037768
12	353	iic119	iic99	5012	5114	0.036120
13	349	iic125x	iic65	4639	2663	0.050194
14	335	iic43	iic65	4328	2663	0.050331
15	328	iic114	iic65	5348	2663	0.042692
16	324	iic125x	iic72	4639	4179	0.038145
17	312	iic125x	iic45	4639	4021	0.037374
18	302	iic45	iic80	4021	3837	0.039968
19	297	iic114	iic125x	5348	4639	0.030650
20	289	iic65	iic72	2663	4179	0.044102

Figure1: The table giving the results of comparing each file with each other, ranked by number of matching trigrams. The column "R" gives the resemblance measure for the two files. The results can be ranked by ordering any of the columns.

When the ranked list of similar pairs of documents appears, the lecturer can select any pair to be examined side by side. Then any suspicious pair of documents can be selected and displayed side by side, with matching text highlighted, as shown in Figure 2.

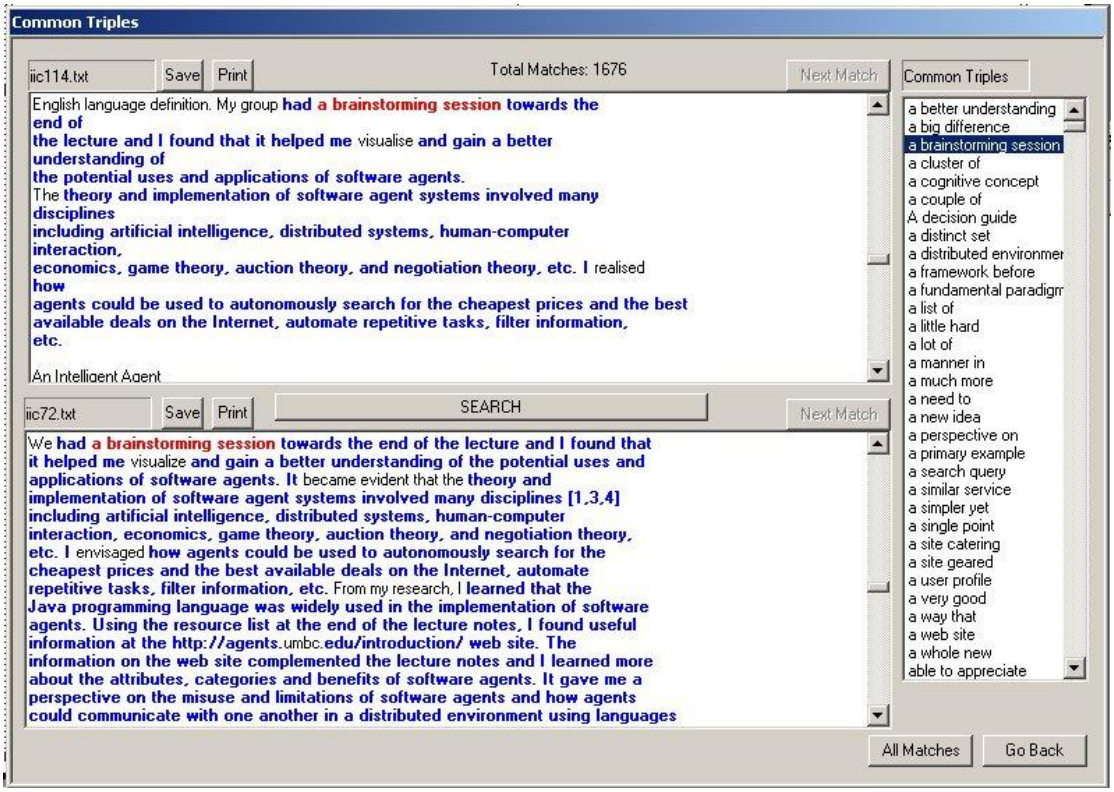


Figure 2: Displaying similar passages side by side. Matching trigrams are bold.

If there is no copying between documents, there will be a sprinkling of matching trigrams. Where plagiarism or collusion has occurred there will be blocks of matching text. Selecting a highlighted word or phrase in one document enables a jump to the matching section in the other document. Texts can be saved, with matching passages highlighted.

The Ferret can be simply installed on any PC with Windows 9x, XP or 2000. On a PC with 750MB RAM Ferret can cope with 320 texts of 10,000 words each; with 250MB RAM, about 200 texts of this size is the limit. It is available at:

`http://homepages.herts.ac.uk/~comqrb1/ferret.zip`

3. Turnitin

The Joint Information Systems Committee (JISC) is supporting a detection service free to Higher and Further Education Institutions from September 2002 to August 2004 [JISC service]. The service uses the Turnitin software from the American company iParadigms [iParadigms]. Electronic comparison of a student's work against Internet sources and all work submitted to their database can be carried out. This will include work submitted by students at other institutions. A JISC sponsored report [Bull 2001] set criteria on which plagiarism detectors were graded on a scale of 1 to 5 on five factors. Turnitin was graded 4 on cut/paste, 4 on "papermills", 5 on collusion, 5 on clarity of reports, 5 on overall feel/user friendliness.

A significant problem with the use of the service is that all work must comply with the Data Protection Act and so students must explicitly give their permission. Advice on the JISC Web-site [Duggan 2003] includes the following:

- all data subjects must provide consent prior to using the service
- it is the responsibility of the institution to ensure consent is given and retained
- data subjects have the right to request removal of data

In the experiments described below the work from UH was submitted anonymously to Turnitin.

3.1 Theoretical basis

The theoretical basis of Turnitin is not known.

3.2 Using Turnitin

Turnitin is accessed through a Web-browser. The lecturer registers with Turnitin, sets up a class homepage with its own enrolment password and creates an assignment. The work can either be submitted by a student using the class enrolment password or submitted by a lecturer/administrator in one of three ways: cut and paste, file upload and bulk upload. The first two methods are suitable when a small number of files are to be checked. All methods are very time consuming: cut and paste and file upload require the author's first and last name to be entered, bulk upload requires the user to browse to the file, open it, type in a unique title (it doesn't use the file name) attach it and then all the files can be submitted at once. A major defect is that the software does not capture the file-name. Duplicate identifiers were allowed.

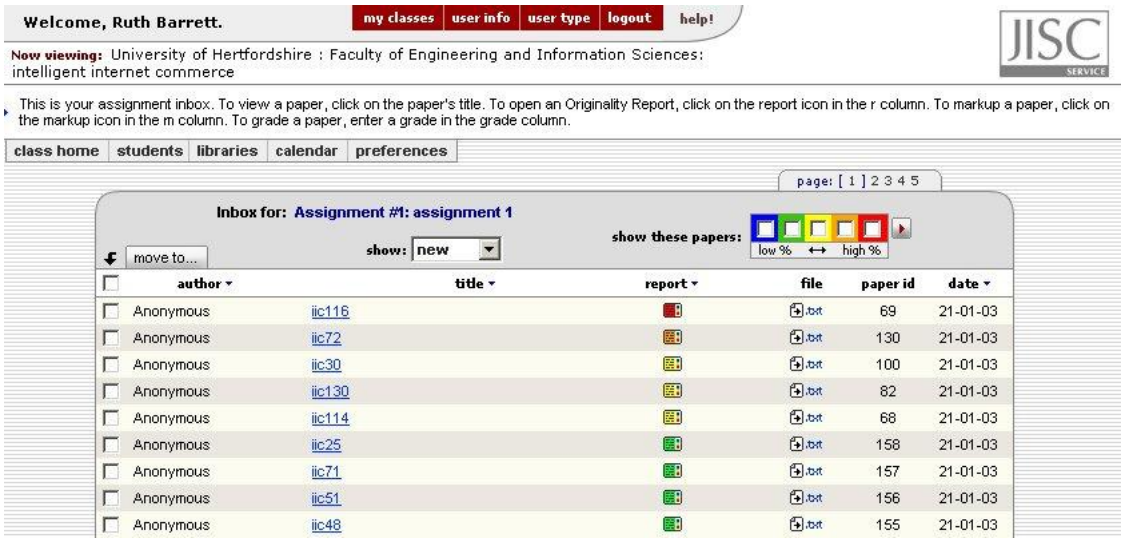


Figure 3: Output from Turnitin on the IIC data from UH

In the version prior to August 2003 it took 35 minutes to submit 105 files, average size 20KB. Each file needed 3 keystrokes: browse, open, attach and it was necessary to scroll the browser box for the last files. The system seemed to get slower after about 30 files and then to hang. Refreshing the screen solved this problem, but this would not be obvious to all users. The August 2003 version is not an improvement; the bulk upload requires 4 fields to be completed. Chandler and Blair [Chandler 2003] also found the bulk upload too time consuming and have written a series of pre and post submission programs. Using these, the students' files are converted to text and concatenated into a single file with a separation point between each student's work. In this way a whole cohort of students is treated by Turnitin as a single submission. At the moment Turnitin is more useful for individual checking of work when plagiarism is suspected than for checking a cohort of students to look for plagiarism or collusion. However, an email from Gillian Rowell on the JISC plagiarism mailing list, 9/10/2003, says "The ability to submit a batch of documents in .zip format will be

available by the end of the year. This will mean that uploading large numbers of papers will be much easier".

Each assignment has an in-box containing the submitted work (Figure 3). The results of the searches are displayed in a "similarity report" for each file within about 10 minutes. This report shows links to Web sources and other files in the database.

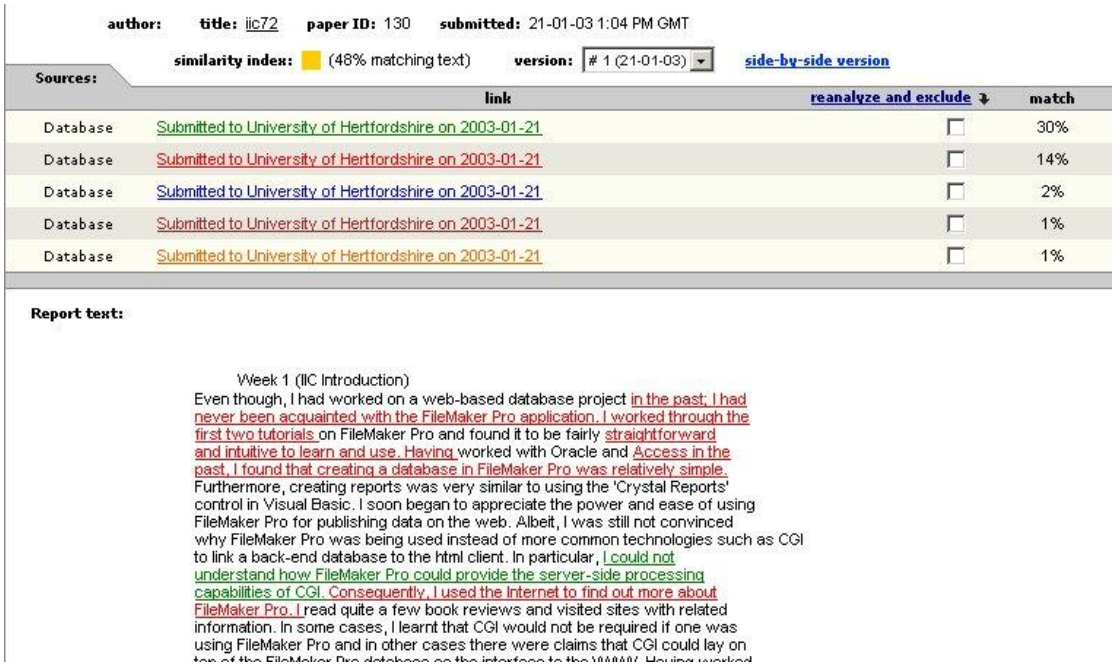


Figure 4: Turnitin's display of a reported text from the IIC data, with % score on the right.

Turnitin ranks the files in order of similarity, displayed as red (high) down to blue (low). This colour code is based on a percentage measure: see Figure 4, and Table 1. This metric seems to be the ratio of number of words in matching passages to all words in one of the files.

4. CopyCatch

In July 2003 a free version of CopyCatch was made available and this is the software used in these trials. This version is an enhanced development of the original CopyCatch which received a grading from JISC of 5 on cut/paste, "papermills", collusion, clarity of reports and overall feel/user friendliness [Bull 2001]. The full commercial version is CopyCatch Gold [CopyCatch].

4.1 Theoretical basis

CopyCatch is not a reliable indicator of plagiarism or collusion. It detects gross copying, but though the measure it produces may be correlated with copying, this is not always the case. It is based on the frequency of matching single words in two documents that are being compared. Obviously, if a word occurs in only one document it has not been copied. Furthermore, if matching words occur a number of times, it is assumed that they are words that would be used frequently without any copying. These may be function words like "and" or "the", or they may be content words expected in a particular topic. However, if a matching pair of words occurs once only, this is taken as an indicator of suspicious similarity. CopyCatch is similar to Ferret in that it exploits the distribution

of words, but because it deals with single words rather than trigrams it has much less discriminatory power. It can be fooled with false positives and false negatives, as described in Section 5.

4.2 Using CopyCatch

CopyCatch software is easily installed on a PC with a Java Runtime Environment. Once installed, a user will be able to click on the icon and the program will run inside Windows. The lecturer chooses the files he wants to check and the system will either compare each file with every other file, or compare each file with just one "comparison" file. The interface is fairly intuitive, but in trials with a number of staff all had difficulty in deciding how to start the comparison. The comparison process also has no progress feedback and so a user will only know this has been completed by repeated clicking until something works.

Files have to be submitted either in Word or text format, as for Ferret. However, if files are not submitted in the correct format they are just overlooked. No report back to the user is made. The system does not say how many files are being processed, so dodgy assignments may escape analysis.

As a preliminary task we ran CopyCatch on a small set of Word test files. Embedded formatting symbols are erroneously taken as "words".

The processed files are ranked in an idiosyncratic form. A "percentage" for each pair is calculated by:

- taking *all* common content words, those shared once and those shared more than once
- taking the percentage of these common words out of all words in each file
- averaging these percentages for the 2 files, weighted for word count.

Pairs of files with a "percentage" score above a chosen threshold are considered likely to contain copied material. Detailed statistics are also given, but these need interpreting carefully and so may not be suitable for infrequent users.

CopyCatch presents results by showing selected pairs of texts side by side, with shared words highlighted, but you have to scroll through to find the correspondence. Similar passages are not displayed side by side as in Ferret. CopyCatch also marks words that are shared just once between documents. This measure is correlated with plagiarism, but is not a reliable indicator.

The limit to the amount of data that can be processed is similar to that of Ferret. CopyCatch can handle 320 texts of 10,000 words each on a PC with 750MB RAM, but only 200 files of this size on a PC with 250MB RAM. Switching applications in the Windows environment while CopyCatch is running will cause CopyCatch to hang.

5. Experiments

Experiments were carried out to determine what levels of similarity indicated suspicions of copying, what degree of copying could be detected, and how effective the systems were in finding plagiarism or collusion in actual students' work.

Data used was as follows.

- IIC (Intelligent Internet Commerce) reports

This course, formerly run on the Computer Science BSc, was chosen for investigation because it was organised in a way that made collusion between students more likely than usual. Students had to work in teams to develop a piece of software, and then produce an individual "Reflective

Commentary" on a weekly basis. They had to switch from working together, sharing disks and programs, to working individually. Three different lecturers marked the work. 124 assignments were examined.

There were 103 pieces of work where the whole Reflective Commentary for each student was saved in one file. The average number of words was about 4,000. There were another 21 pieces of work where the Commentary was saved in sets of smaller html files, 281 in all. Each of these averaged about 400 text words.

- The Federalist papers

These very well known essays, on which the American constitution was based, have been closely examined for over two centuries. We have selected 51 by one author (Hamilton), many of them on similar subjects. (For these essays there is no dispute over authorship). As essays by the same author on similar subjects they are likely to exhibit a high level of similar characteristics, so they are useful data for setting bench marks to see what is the highest likely level of Resemblance in independently written texts. The length of these documents was from 1000 – 5,500 words.

- Gutenberg data

This is data taken from the Gutenberg site, where classical texts are available in electronic form. In order to simulate large quantities of data, such as the 320 essays of 10,000 words each from the Staff College field trial (see below), texts were taken and divided into files of 10,000 words each. Then pseudo plagiarised files were created by cutting and pasting 100 / 200 / 300 / 400 words from one file into another. This was again used for setting bench marks and for seeing at what length a copied passage could be detected.

5.1 Establishing baselines

Using Ferret, we carried out the following experiments. After processing a set of files and obtaining a similarity ranking, pairs of files can then be compared side by side. As this is done any copying can be seen as blocks of largely highlighted text, a subjective decision can be made as to whether there has been any plagiarism, and the associated R (Resemblance) measure noted.

5.1.1 Analysing the IIC data with Ferret

Matching sections of text that were judged to be plagiarised were found in 14 out of the 124 pieces of work. Two of these were sizeable chunks; most were matching sentences sprinkled through the work. As illustrated in Figure 5, in the files with matching texts, almost all had matches with more than one other file.

Student 1
In the lecture we looked at the 1998 data protection act, I felt that I already had good knowledge of this subject area from my business issues course but soon realised that the act was a lot more difficult to apply to the internet, as it seems that many sites are able to pull information about you as you access their sites without your permission.

Student 2
In the lecture we looked at the 1998 Data Protection Act, I felt that I already had good knowledge of this subject area from my Industrial Placement but soon realized that the act was a lot more difficult to apply to the internet, as it seems that many sites are able to pull information about you as you access their sites without your permission.

Student 3
In this lecture we looked at the 1998 data protection act, I felt that I already

had good knowledge of this subject area from this course. But soon realised that the act was a lot more difficult to apply to the internet, as it seems that many sites are able to pull information about you as you access their sites without your permission.

Figure 5: These three examples illustrate students copying from each other. Text that is common to all is underlined.

As the detector analyses word patterns, it can work on html as well as ordinary text. Non-alphabetic symbols are ignored. Parts of embedded html commands are left as "words" in the text, but this could be viewed as background noise, and does not undermine the operation.

Each piece of work was compared with every other piece. The html files were left as they were. Thus, there were 103 + 281 files to compare, 73536 comparisons. On the 750MB RAM machine the stage 1 and stage 2 processing takes about 3 minutes. Core stage 2 processing takes less than a minute. The third stage, when files are compared side by side, takes as long as the lecturer chooses.

Documents from the Web can be added to the set processed by Ferret to allow some copying from the Internet to be detected, see Figure 6.

Web text
In 1998, just 15 percent of online households felt safe using their credit cards for online purchases. In the most recent study, this number rose to 53 percent. Furthermore, the percentage of consumers who feel that the Internet is not a secure place to use a credit card has plummeted from 71 to 40 percent. The survey also found that the longer consumers are online, the more comfortable they become with credit card security.

Student 4
In 1998, just 15 % of online users felt safe using their credit cards for online shopping [3]. In the most recent study, this number rose to 53%. The survey also found that the longer customers are online, the more comfortable they become with credit card security.

Student 5
In 1998, just 15% of online users felt safe using their credit cards for online shopping. Most recently number reached to 53%. The research also states that the longer customers come online, they become more comfortable with credit card security.

Figure 6: In these examples, matches with the Web are underlined.

Results
Top ranking pairs of texts are shown in Figure 1, and some are shown in Table 1. Where R >0.04 copying was found, below 0.04 it was usually absent. This threshold is equivalent to about 8% of matching trigrams as a proportion of all trigrams in a text.

Only two sizable pieces of matching texts were found: one of 8 paragraphs, one of 6. Other matches were of sentences sprinkled through the texts, from 2 to 10 sentences. Sometimes there was a cluster of 3 or 4 sentences, other times they were just singletons. Some sentences were well written, probably from the Web; others were poorly written, copied from student to student.

However, in files 11 and 25, R = 0.02, and 5 matching lines were not detected until a random choice of file pairs was examined. The matching trigrams accounted for 3% and 6% of the total number of trigrams in each file.

File numbers	**Plagiarised** (subjective judgement)	**Ferret** R metric Resemblance	% trigrams matching	**CopyCatch** similarity	**Turnitin** "similarity" for each file
72, 114	Yes	0.21	40%, 31%,	78%	30%, 29%
30, 43	Yes	0.14	30%, 22%	68%	27%, 20%.
72, 88	Yes	0.13	21%, 24%	73%	14%, 14%
47, 114	Yes	0.04	8%, 7%	65%	3%, na
30, 72	No	0.03	7%, 5%	63%	na, na
65, 88	No	0.02	5%, 4%	63%	na, na
114, 43	No	0.02	4%, 4%	61%	na, na
25, 11	No? 5 matching lines	0.02	6%, 3%	58%	3%, 1%

Table 1: Excerpt from results on IIC data from the three systems. It shows that all three identify gross copying, but some independently written texts score over 60% similarity by CopyCatch

5.1.2 Analysing the IIC data with Turnitin

Turnitin ranks the files on "percentage" matches from low (blue) to red (high). Six files were identified for further investigation. One had a 97% match with a www link, which turned out to be the same report put up by the student on his own Web page. Another had 19% similarity with 15 different www sites: this piece of work was an example of cut-and-paste plagiarism. The other four files (see 72, 30, 114 and 25 in Table 1) identified potential plagiarism or collusion within the set of files.

There was no efficient way of checking the 55 files at the next level of similarity.

Comparing these results to the Ferret we see that Turnitin detected the same worst cases.

5.1.3 Analysing the IIC data with CopyCatch

CopyCatch was also run on the IIC data, see Table 1. Gross plagiarism was detected, as in the other two systems. However, when we look at the 10 pairs of files that CopyCatch displayed at the top of its similarity ranking, four of them (52 and 53, 45 and 69, 49 and 80, 28 and 88) were not judged to have plagiarised passages, though they had a high number of single words in common. Other pairs of files that *were* judged to include plagiarised passages were lower down the ranking. Depending on where a threshold was set, false positives or false negatives, or both, were produced.

Essay numbers	**Significant matching passages?** (subjective judgement)	**Ferret** R (Resemblance) metric	**CopyCatch** "similarity"
06, 74-edited	Yes. Paragraph from 06 pasted into 74	0.06	31%
81, 82	Borderline 19 word quote match	0.04	50%
67, 76	Yes ? 35 word quote match	0.04	37%
81, 83	Borderline 19 word quote match	0.03	60%
32, 33	No 17 word title match	0.03	45%
80, 82	No 17 word title match	0.03	47%
69, 74	No	0.03	34%
80, 81	No	0.03	53%

Table 2: Excerpt from results of analysing Federalist papers by Ferret and CopyCatch, in descending order of R (Resemblance) metric.

5.2 Analysing the Federalist Papers

This data brings out the fact that determining whether copying has taken place is ultimately a subjective judgement, and the boundary is a grey band rather than a black and white division. "Copying" between these essays is either the use of the same quotation, or else the same title.

The first entry Table 2 records the result of processing 2 files, where a paragraph from one essay has been pasted into another. Ferret detected this, but CopyCatch did not. On the other hand CopyCatch produced a fairly high similarity measure for files with borderline matching.

For documents of about this size, an R score of about 0.04 with Ferret is seen empirically to be an appropriate threshold, below which texts can be sensibly judged as independently written.

This experiment could not be done with Turnitin, as these well known essays would be bound to throw up many matching texts.

5.3 Analyzing the Gutenberg data

The purpose of these experiments was to simulate the processing of the Staff College data (see section 6.2), where field trials were planned. Classical texts from the site were divided into 10,000 word sections. Again, these experiments could not be carried out with Turnitin, as these well known texts would have thrown up many matches with copies on the Web.

First, it was determined that 320 essays of 10,000 words each could be processed. This could be done by both Ferret and CopyCatch on a PC with 750MB RAM. It could not be done on a PC with 250MB RAM: in both cases the process ground to a halt.

Secondly, we wanted to establish base lines: what was the highest Resemblance measure for these texts?

Thirdly, we wanted to test what degree of matching could be detected, so some texts were doctored by copying and pasting passages of varying sizes from one text to another.

Results

For comparisons of passages from the same book the highest R measure was 0.03 (for passages from "The decline and fall of the Roman empire".) For passages from different books, the R measure was much lower, down to 0.002. This reinforced the view that R = 0.04 is an appropriate threshold below which texts can be considered independently written.

The CopyCatch similarity measure was as high as 60% for texts from the same book. On inspection, there was not judged to be significant matching of passages, though there were many matching single words.

Texts were doctored by copying and pasting passages of 100 / 200 / 300 / 400 words from one file to another, see Table 3. Ferret could detect copies of 300 or 400 words, but if only 100 or 200 words had been copied the R measure was sufficiently low that it could have occurred without copying. The CopyCatch similarity measure could not detect the doctored texts.

File names	**Copied text pasted in?**	**Ferret R** (resemblance) metric	**CopyCatch** "similarity"
dnfreaw dnfreax	None	0.03 (highest score)	62% (highest score)
dnfreaa dnfrebu	None	0.01	43%
zdnfreaa dnfrebu	300 words	0.03	46%
zdnfrebn dnfrebu	300 words	0.04	57%
zdnfrebl dnfrebw	400 words	0.04	50%

Table 3: Excerpt from results of analysing Gutenberg data by Ferret and CopyCatch

6. Field Trials

The Ferret has undergone field trial at Maastricht University and at the Joint Services Command Staff College.

6.1 Field Trials at Maastricht University

The Ferret has been developed using the English language. But the developers were interested whether the positive findings with the program could be replicated with papers in another language, so when Maastricht University contacted them to ask if they could try out the Ferret, permission was readily granted, on condition that they shared their experience.

At the Maastricht University three faculties were involved in the Ferret trials: Law, Health Sciences, and Psychology. In the first year students have to write a paper of five to six pages on a certain topic, the topic depending on the subject area. The number of papers submitted was Law: 256, Health Sciences: 275, and Psychology: 31. The program ran smoothly in all three runs and it has worked equally well in Dutch as in English.

The software can only guide the teacher to potential plagiarism; then academic judgement must be applied. After the teachers studied the paired papers, they decided that four pairs resembled each other to such a degree that plagiarism was assumed. After consulting the paper writers, the teachers decided that in three cases plagiarism was found. Six students from the Faculty of Law were penalized according to the University rules and regulations. The teachers involved were impressed by the user-friendliness and the speed of the output, and also produced some recommendations for improving the interface. These have been implemented in a newer version of the program.

6.2 Field Trials at the Joint Services Command Staff College

This work comprised 323 files, total size 114Mb, each piece of work averaged 10,000 words. The work was also submitted to Turnitin by JSCSC administrative staff.

With Ferret, the preliminary conversion to text process presented unforeseen difficulties. Of the 323 files, 12 files needed renaming, either to remove full-stops from file names or to add the suffix .doc. 38 files had a suffix .obd (several documents bound into one). Ferret converted these files to one text file but the size increased considerably, and embedded formatting symbols were retained.

The plagiarism processor was run on these files on a 130KB machine, but did not complete as the memory was insufficient. This was repeated on a 1GB RAM machine. This crashed at 73% with message from the Windows system. The largest of the files that were generated from the original .obd files were removed and the remaining 290 files took 67 seconds to process.

Display of results: All files that were originally in .obd format were at the top of the matches because the garbage added in the conversion to text by Microsoft Word was identical in each file. All files that started as .obd were removed and the plagiarism detector re-run. The top two file pairs (R values 0.032 and 0.026) were checked and no obvious collusion was found. This opinion was confirmed by staff at JSCSC. The work involved in producing a file of the assessed part of each student's work from the .obd files was not felt to be worthwhile for this trial.

Conclusion: On a machine of 1GB Ferret is very fast in both the collusion detection and reporting stages. The difficulties are with file naming and file formats.

On the submission to Turnitin by JSCSC staff it was reported that there were about 10 worrying examples of plagiarism from the Web, of which 4 were followed up with interviews before

sanctions were decided upon. The staff were pleased (a) with the deterrent effect on the 330 students and (b) the speed and ease of getting and analysing the results.

7. Conclusion and Future Work

Ferret has been enhanced, by adding a facility to automatically include in the processing relevant documents found on the Web. The lecturer has to type in keywords, and the processor will return the top 50 -100 retrievals, automatically convert them to text and add them to the folder holding other files to be processed. This is a move towards widening the scope of Ferret, but it will not be able to compete with Turnitin's vast database of documents.

Ferret's advantages are that it performs effectively, detecting collusion and copying between students, and to some extent detecting copying from the Web. It is an easy to use, local system. The types of files it can accept are .doc, .rtf, .txt, .pdf (shortly), and we plan to cater for other file types.

Turnitin is a commercial product that charges for its service – paid by JISC only until August 2004.

Appendix A, Evaluation of three plagiarism detectors, gives a summary of our conclusions in a tabular form.

Further work includes improving the Ferret interface, and enhancing the Web searching facility. We may also work on accommodating .odb files. We hope to run more field trials.

The Ferret has also been used in information retrieval, for one of the TREC (Text Retrieval Competition) tasks, and we expect to expand its use in this field.

References

[Broder 1998]. Broder, A. Z., *On the resemblance and containment of documents*, Compression and Complexity of Sequences, IEEE Computer Society 1998.

[Bull 2001] Bull, J. *et al*., *Technical Review of Plagiarism Detection Software*, Report, Computer-Assisted Assessment Centre, University of Luton 2001.

[Chandler 2003] Chandler, A. & Blair, L. *Batch Plagiarism Detection with Turnitin*, <http://www.comp.lancs.ac.uk/computing/users/angie/plagiarism/batch/Turnitin.htm> accessed Jan 2003

[CopyCatch] *CopyCatch Gold*, CFL Software Development, <http://www.copycatchgold.com>

[Duggan 2003] Duggan F., *Advice and Guidance: Data Protection*, JISC 2003 <http://online.northumbria.ac.uk/faculties/art/information_studies/Imri/Jiscpas/docs/northumbria/Fiona_workshop.ppt>

[Gibbon *et al*. 1997]. Gibbon D., Moore R., Winski R., *Handbook of Standards and Resources for Spoken Language Systems*, Mouton de Gruyter 1997.

[iParadigms]. iParadigms, <http://www.iparadigms.com>

[JISC service] JISC plagiarism detection service, <http://www.submit.ac.uk>

[Lyon *et al*. 2001] Lyon C, Malcolm J, and Dickerson B, *Detecting short passages of similar text in large document collections*, Proceedings of EMNLP (Empirical Methods in Natural Language Processing) 2001.

[Lyon *et al*. 2002] Lyon C, Malcolm J, and Dickerson B, *Incremental retrieval of documents relevant to a topic* Proceedings of TREC (NIST/DARPA sponsored Text Retrieval Competition) 2002.

[Manning 1999] Manning C. D., and Schutze H., *Foundations of Statistical Natural Language Processing*, MIT 1999.

[Ney *et al*. 1991] Ney H., Martin S., Wessel F., *Statistical Language Modelling using leaving-one-out.* In S Young and G Bloothooft, editors, Corpus Based Methods in Language and Speech Processing. Kluwer Academic Publishers 1997.